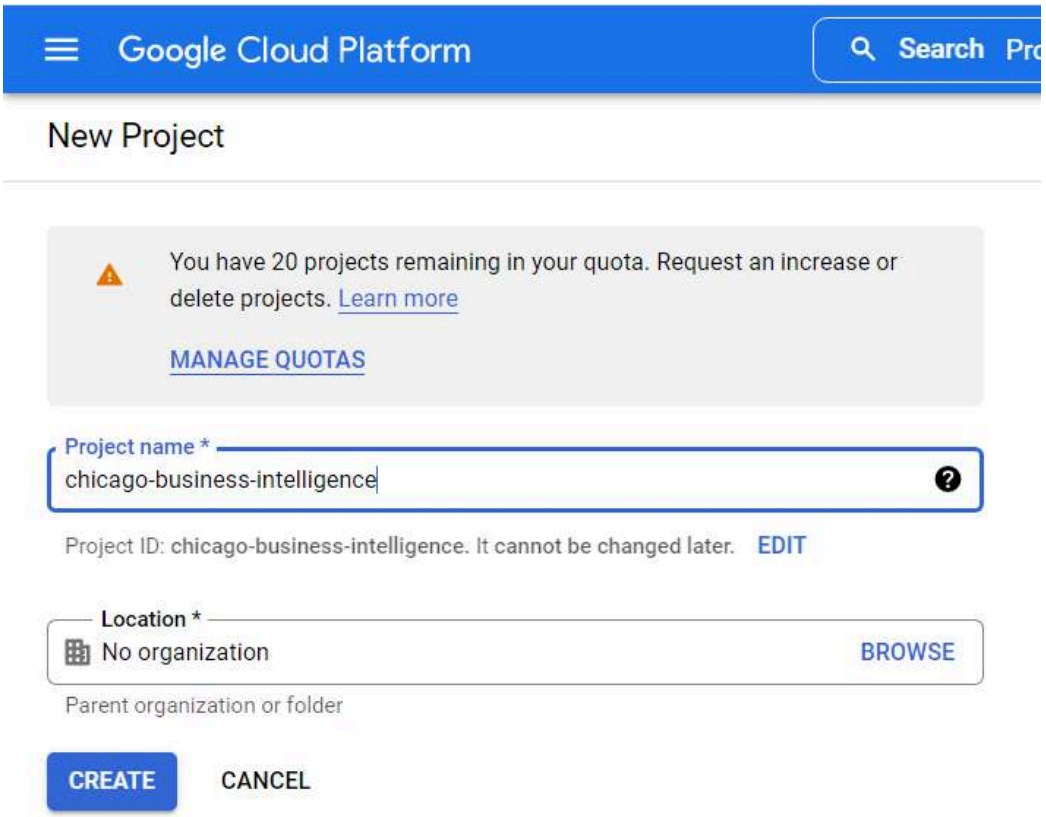
Deploying Go Microservice for Chicago Business Intelligence on GCP using SQL, Cloud Run, CI/CD Triggers for GitHub Repo

Step1: Initial Setup for Google Cloud Platform

- Install the google cloud CLI on your local machine.
- Create a new project on your google cloud console. Make a note of the project id and project Name.

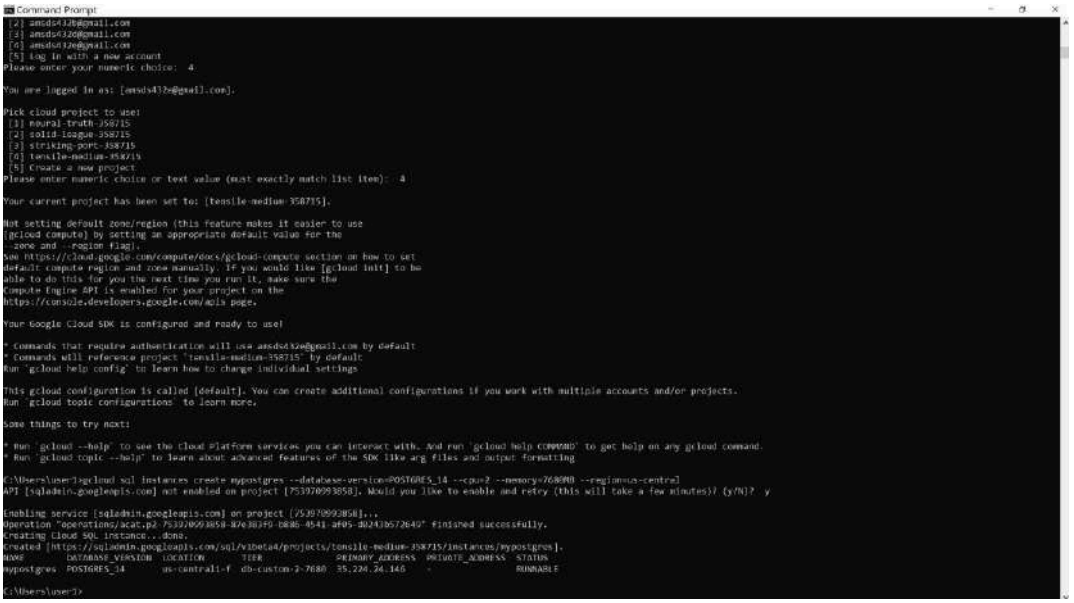


- After creating a project on Google Cloud Console execute "**gcloud init**" command on your local machine and select the project created above when prompted.

```
Your current project has been set to: [chicago-business-intelligence].
```

Step 2: Postgres database Setup

- Create a database instance of postgres using the following command.
  "**gcloud sql instances create mypostgres --database-version=POSTGRES_14 --cpu=2 --memory=7680MB --region=us-central**"



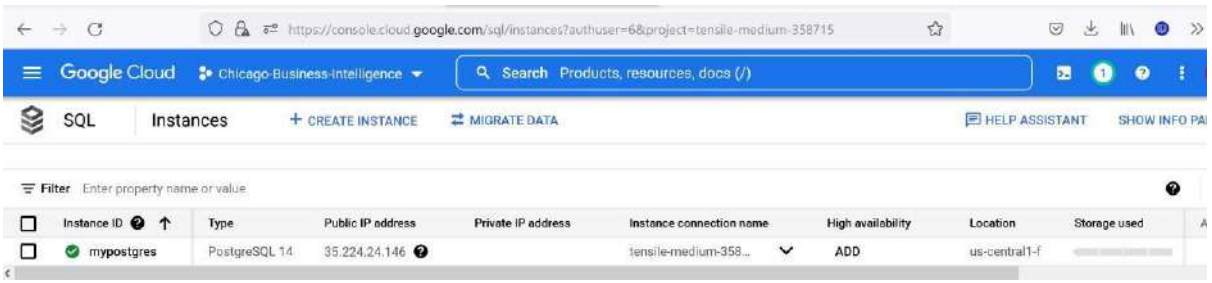
- Create sql users on the database instance using the following command.
  "gcloud sql users set-password postgres --instance=mypostgres --password=root"

```
C:\Users\user1>gcloud sql instances create mypostgres --database-version=POSTGRES_14 --cpu=2 --memory=7680MB --region=us-central
API [sqladmin.googleapis.com] not enabled on project [753970993858]. Would you like to enable and retry (this will take a few minutes)? (y/N)?  y

Enabling service [sqladmin.googleapis.com] on project [753970993858]...
Operation "operations/acat.p2-753970993858-87e383f9-b886-4541-af05-d0243b572649" finished successfully.
Creating Cloud SQL instance...done.
Created [https://sqladmin.googleapis.com/sql/v1beta4/projects/tensile-medium-358715/instances/mypostgres].
NAME        DATABASE_VERSION  LOCATION       TIER              PRIMARY_ADDRESS  PRIVATE_ADDRESS  STATUS
mypostgres  POSTGRES_14       us-central1-f  db-custom-2-7680  35.224.24.146    -                RUNNABLE

C:\Users\user1>
```

- Create a database for our microservice using the following command.
  "gcloud sql databases create chicago_business_intelligence --instance=mypostgres"

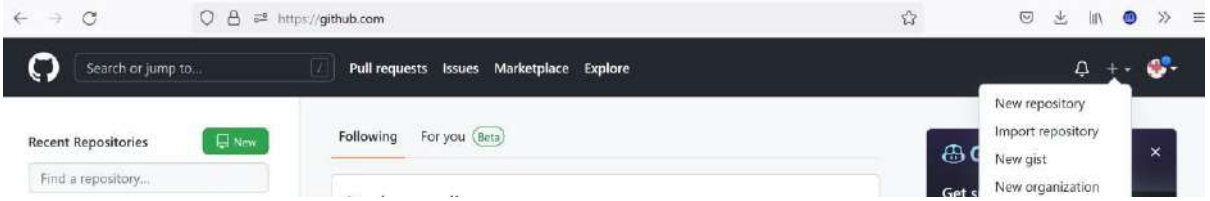
```
C:\Users\user1>gcloud sql databases create chicago_business_intelligence --instance=mypostgres
Creating Cloud SQL database...done.
Created database [chicago_business_intelligence].
instance: mypostgres
name: chicago_business_intelligence
project: tensile-medium-358715

C:\Users\user1>
```

- Open Google Cloud console, search for SQL and confirm that the database instance is up and running.

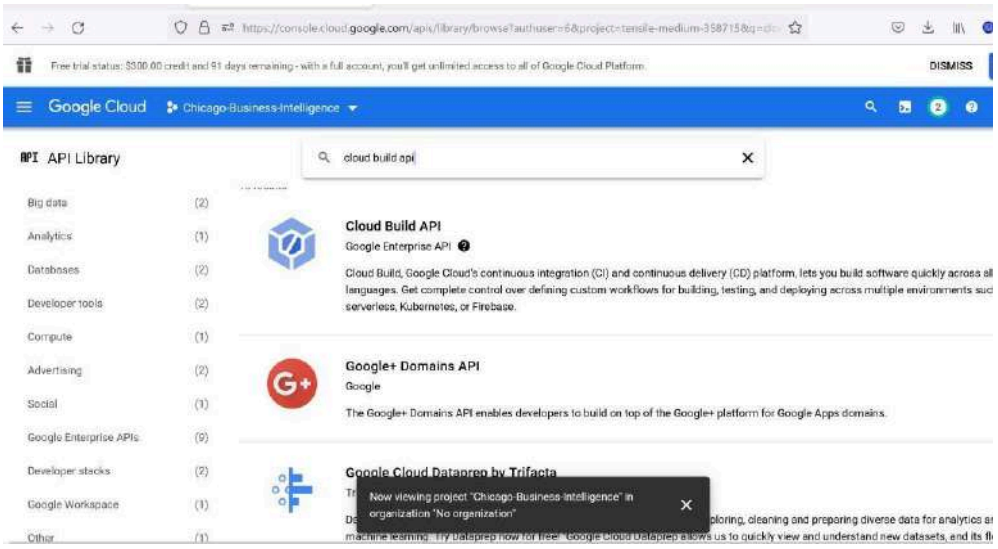


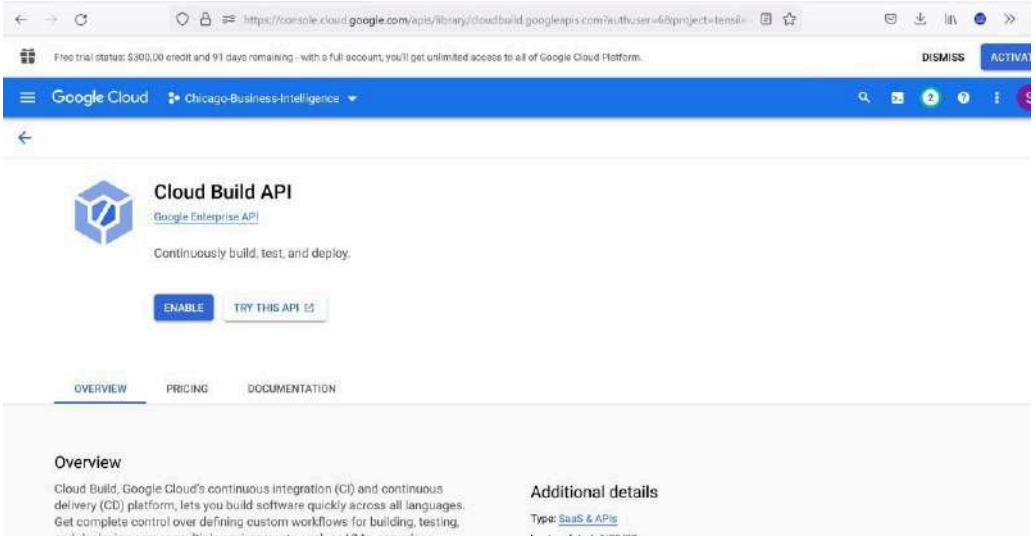
Step 3: Setting up continuous deployment using cloud build.

- Create a **Repository** on **GitHub** to store the source code for your CBI project.

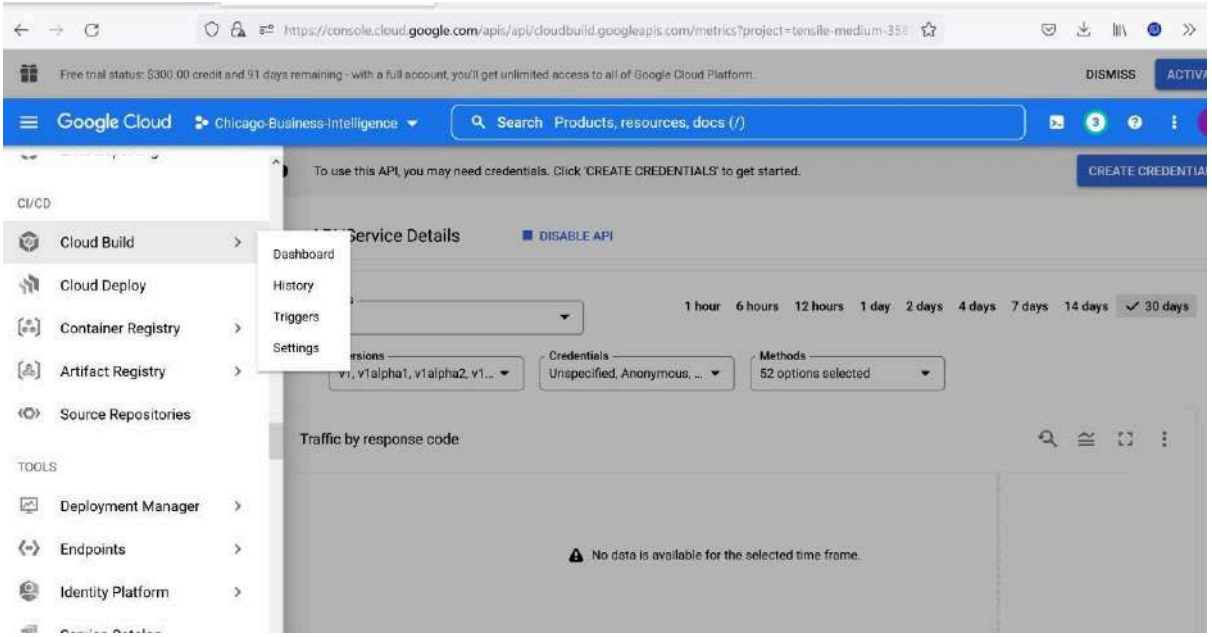


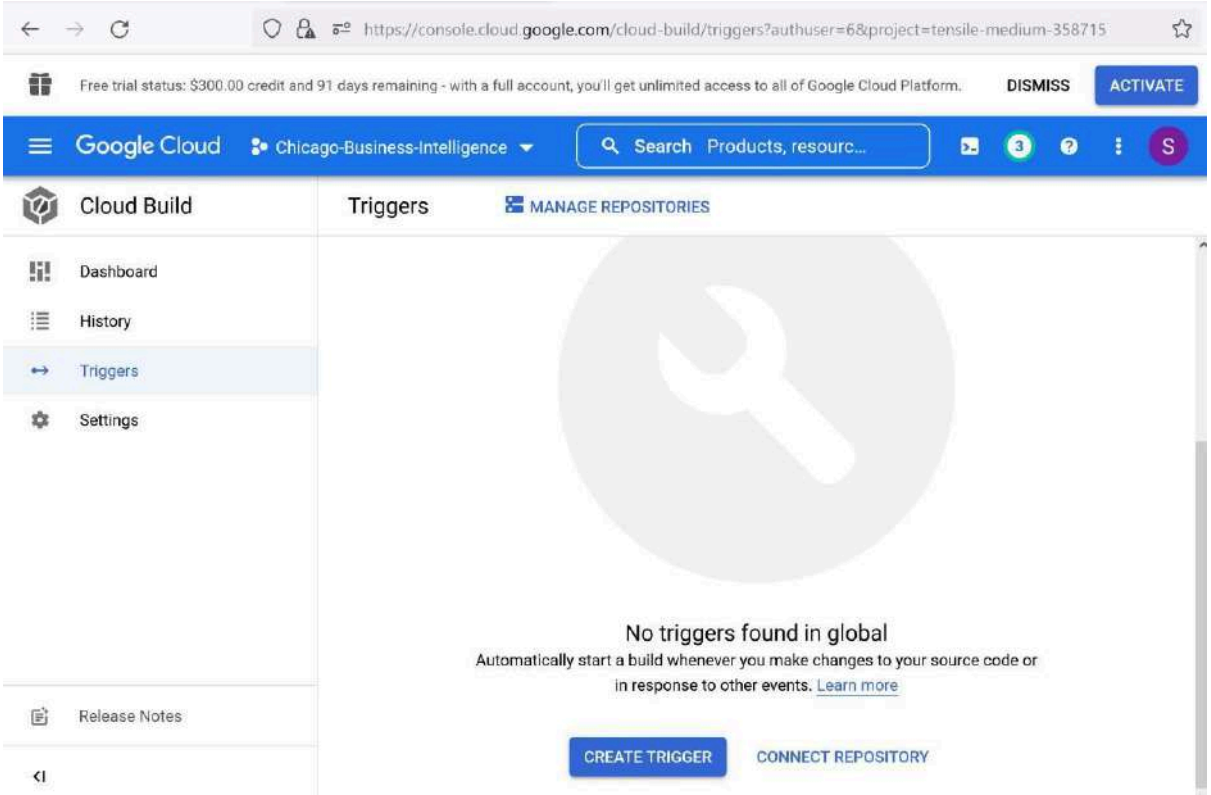
- Open Google Cloud Console, Search for Cloud build API and Enable it for your project.



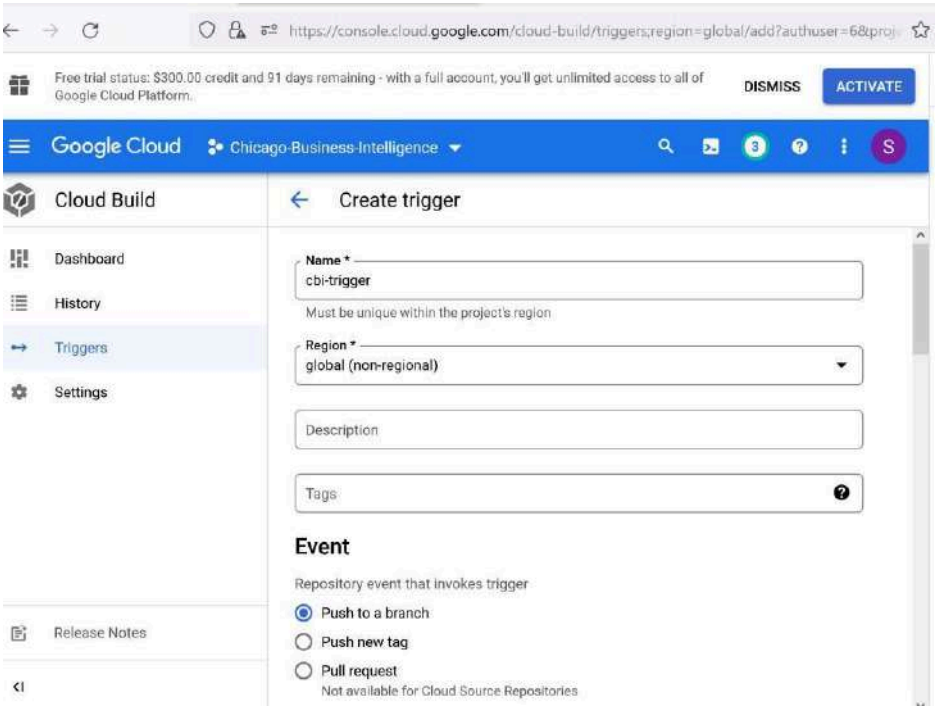


- After the API is enabled, click on the create trigger button.

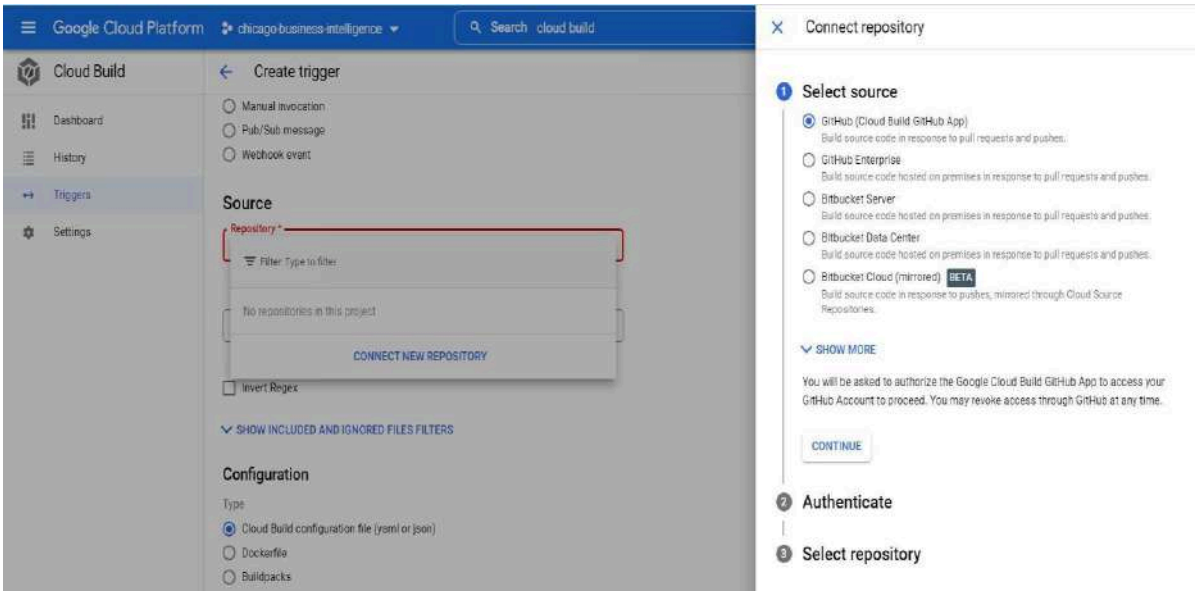




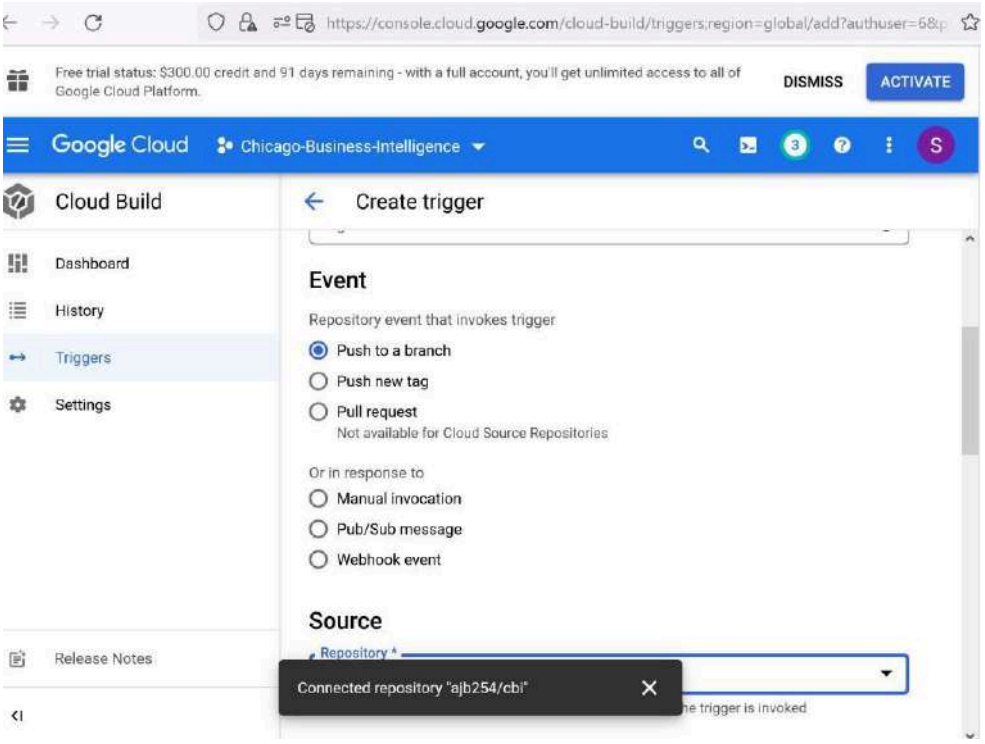
- Fill the details for the trigger as shown in the below images.



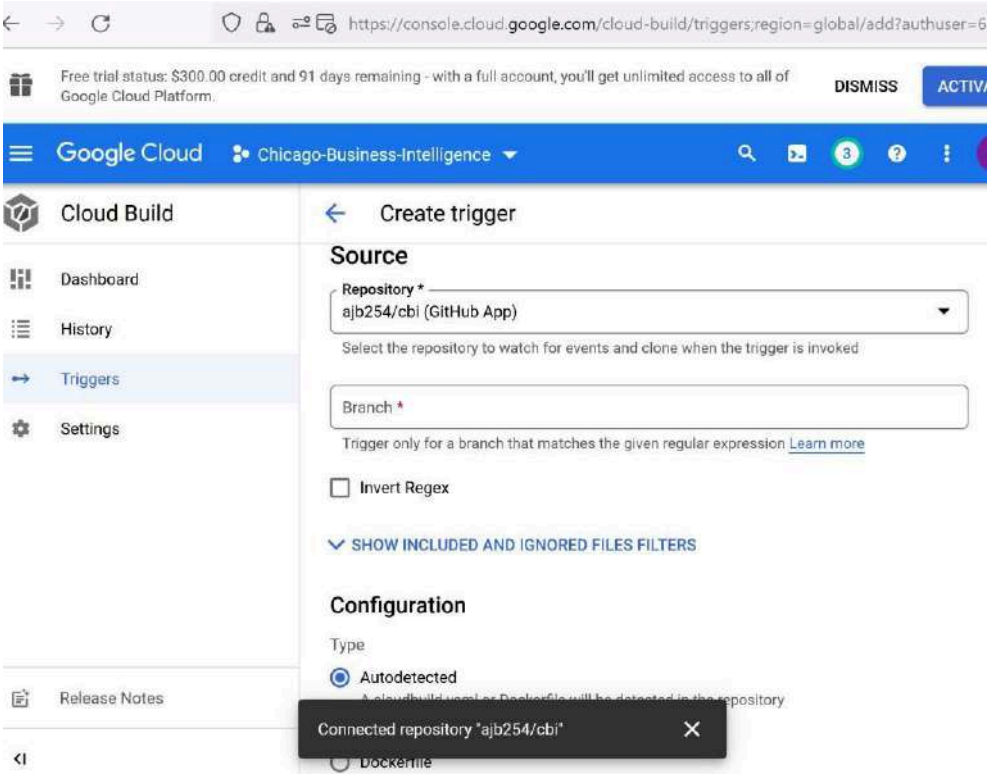
- Click on connect repository, select GitHub and authenticate.



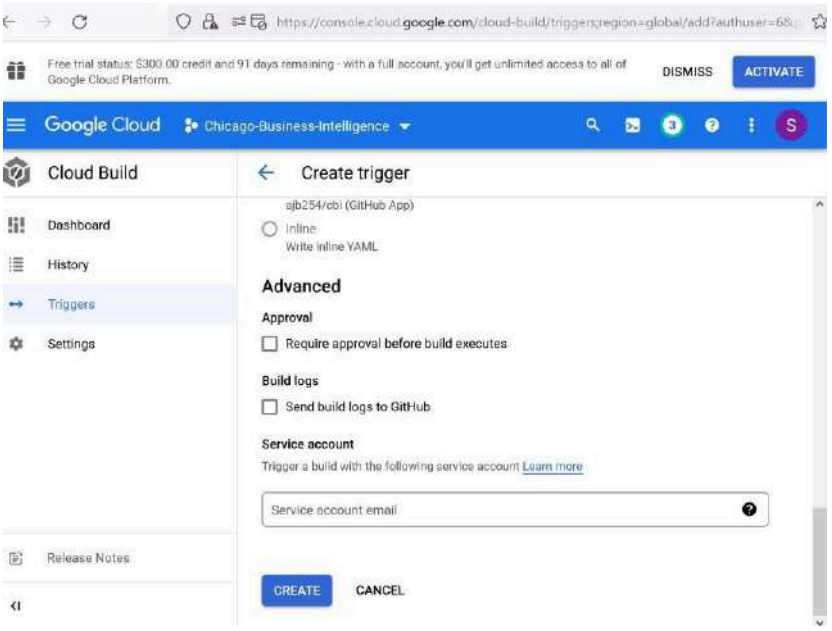
- After authentication select the repository created for Chicago business intelligence.



- Select the repository after connecting the project.

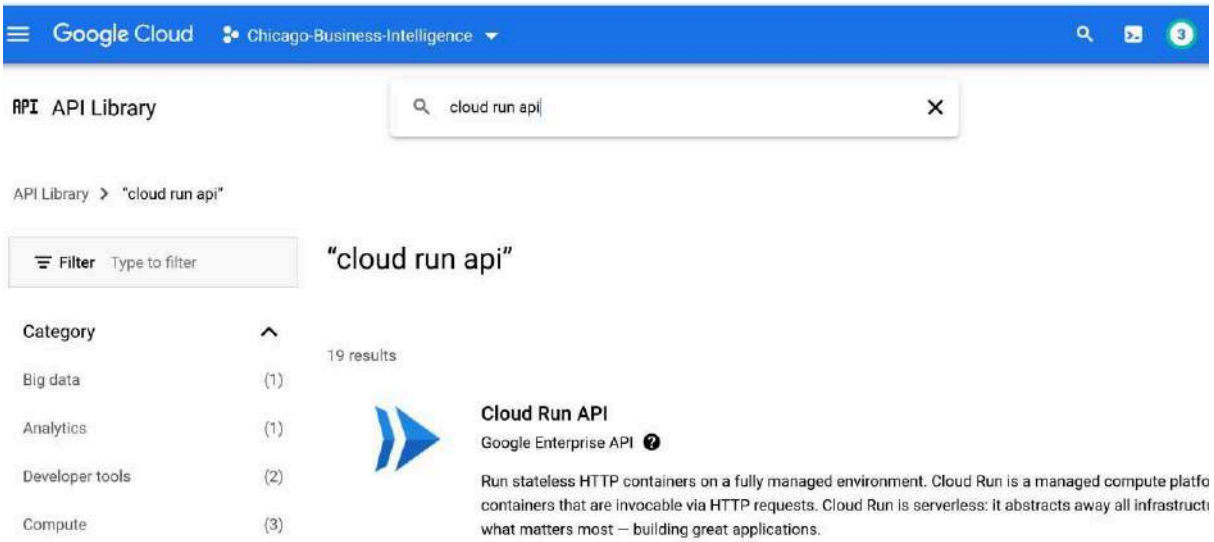


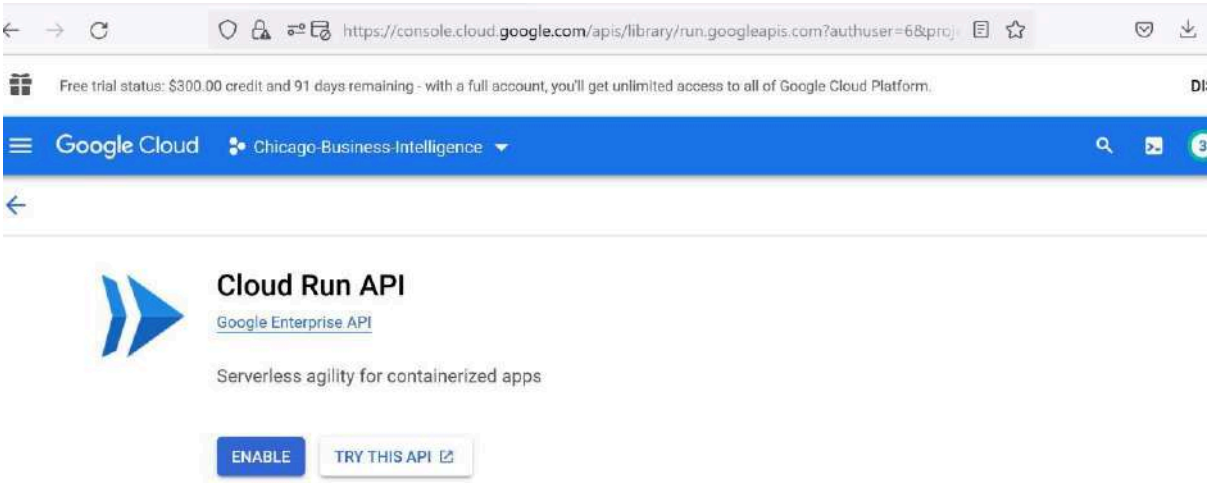
- Click on create to create the trigger.



Step 4: Setting up the containers for Go-microservice and Pgadmin

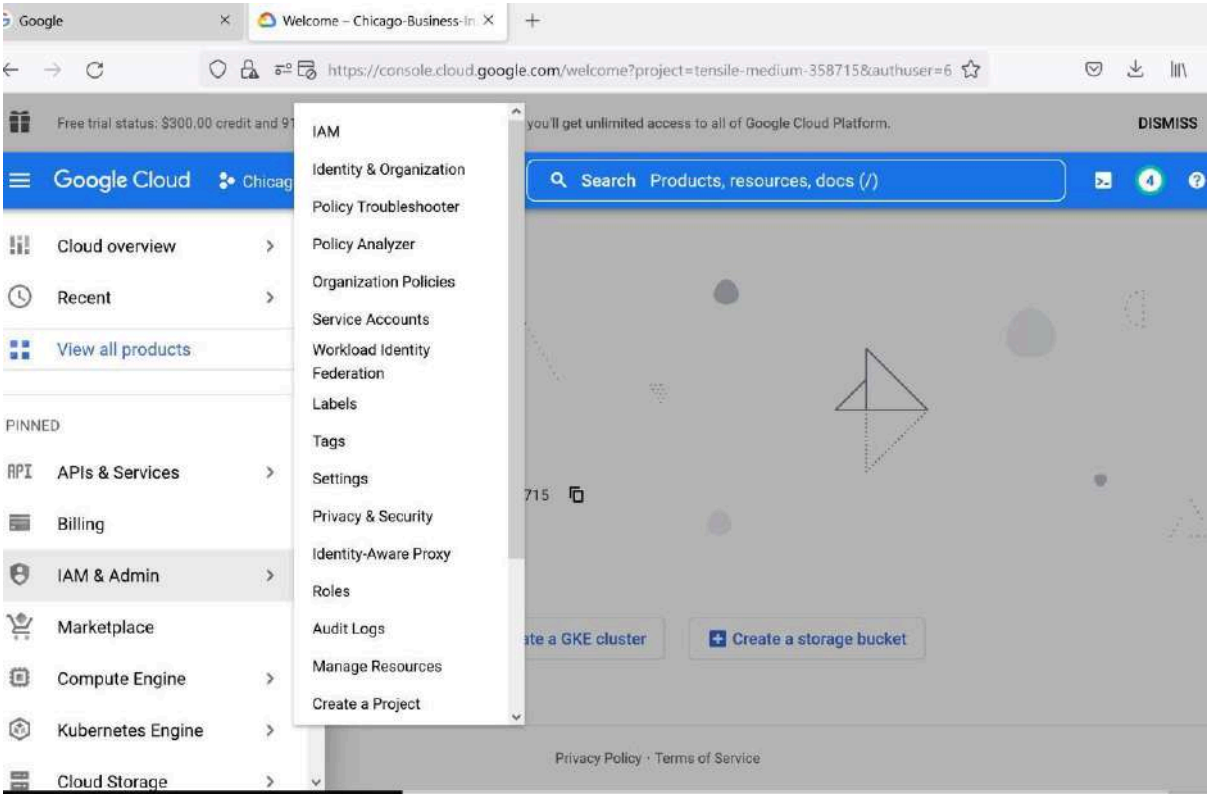
- Enable cloud run api for your project.

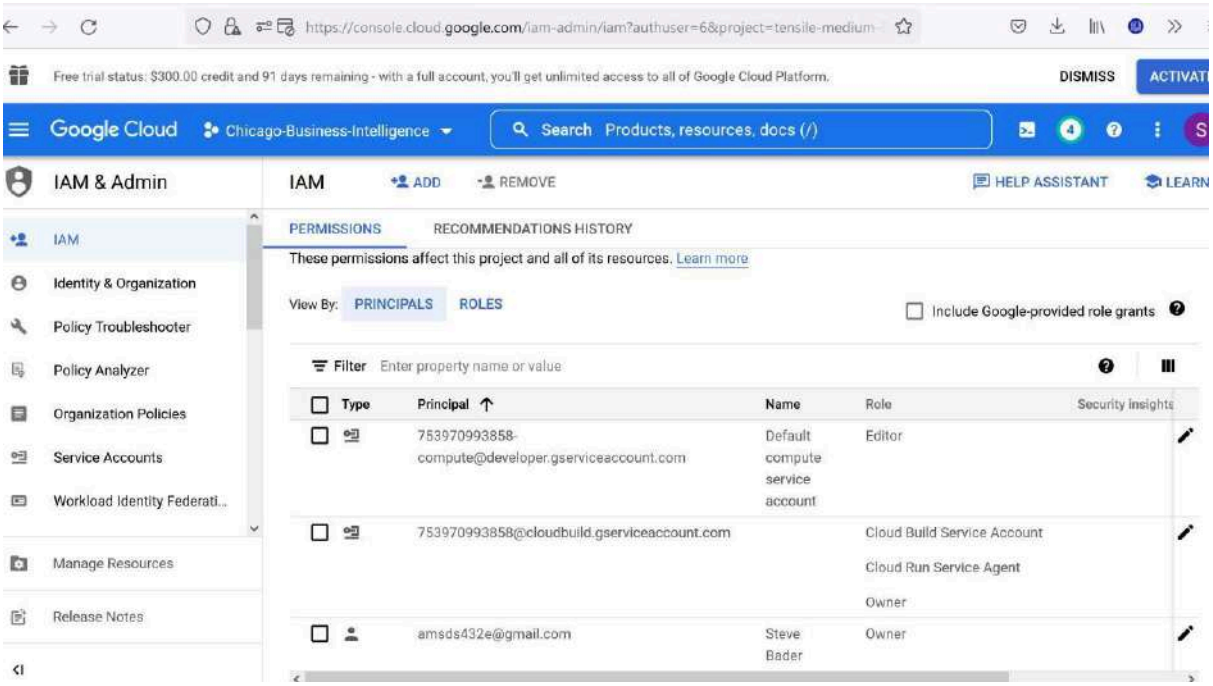




Step 5: Enable IAM permissions/roles

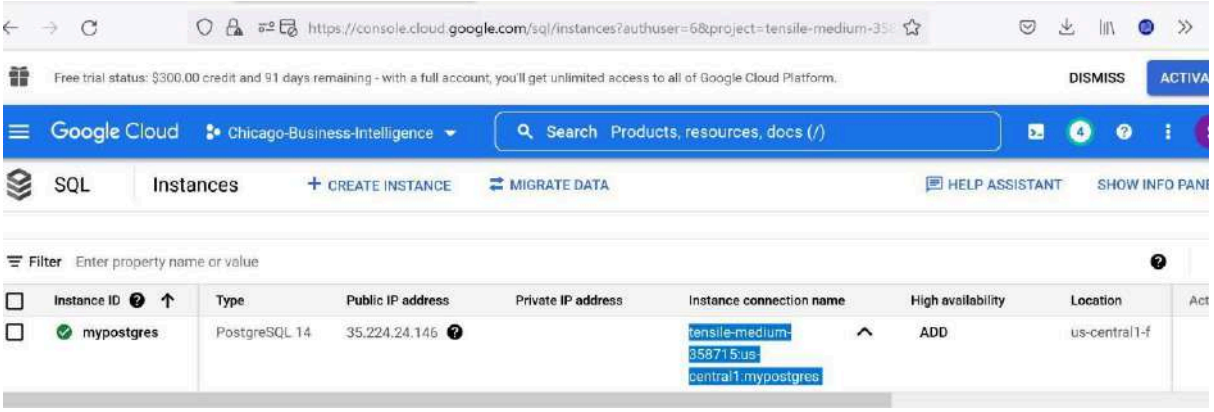
- Go to the IAM page and make sure all the required roles are enabled for the project.





Step 6: Get the Postgres DB instance connection name

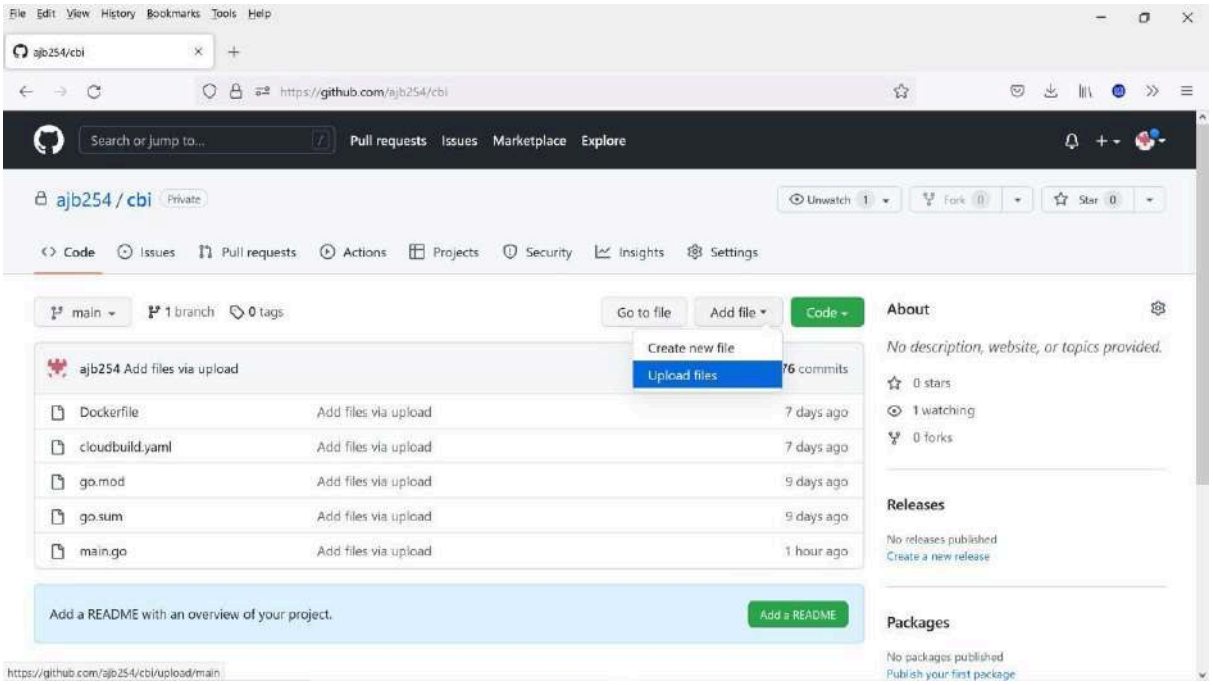
- The images for all the microservice are created with the help of cloudbuild.yaml file.
- Go to the postgres instance created in the previous steps and copy the instance connection name.



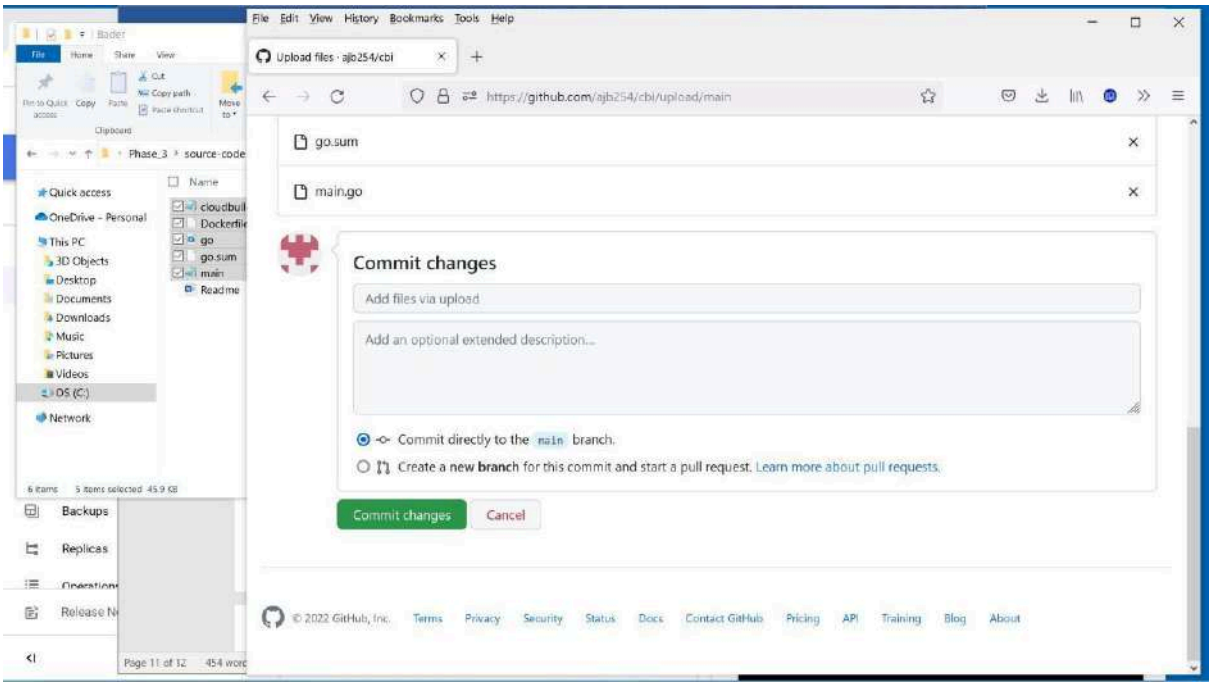
- Go to line "**.env**" file and update the POSTGRES_HOST name string with your instance connection name as shown below. Example is shown below.

  `.env`
  POSTGRES_HOST="/cloudsql/chicago- business-intelligence:us-central1:mypostgres"

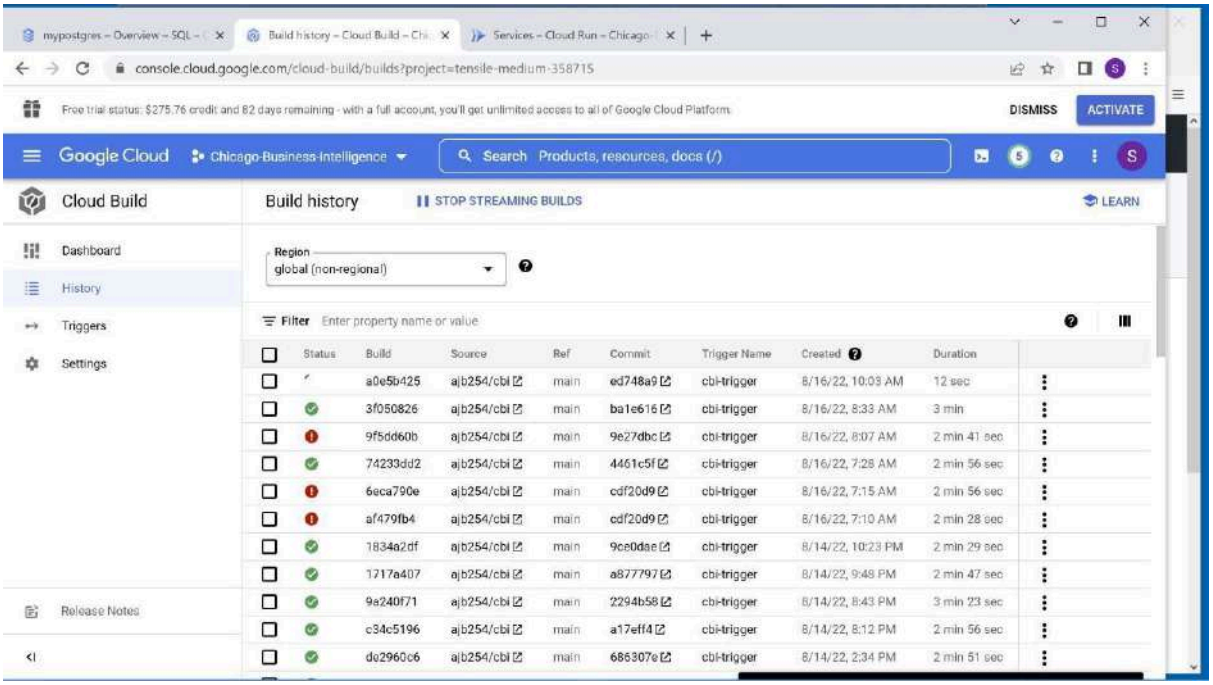
- Push the source code along with the cloudbuild.yaml file to the GitHub repository created in prior steps.



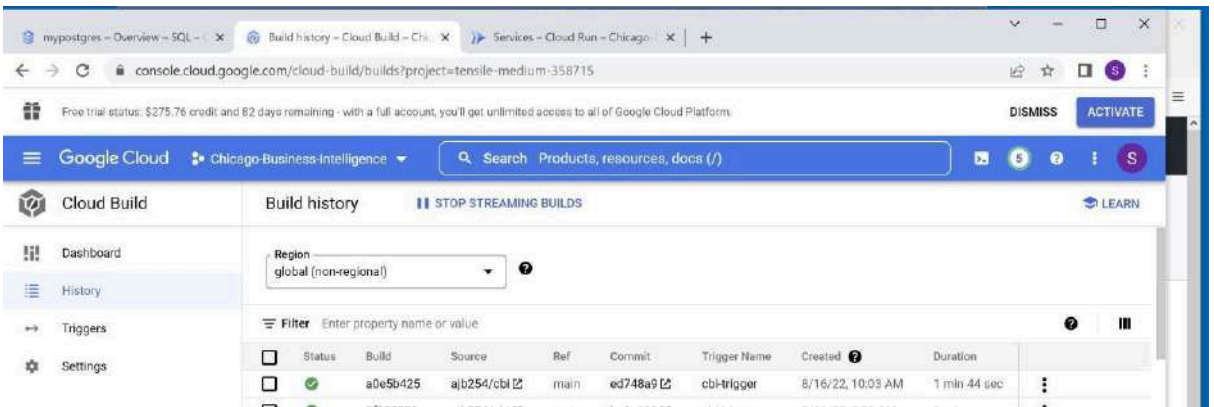
Step 7: Push commits to GitHub, run trigger, and view microservices

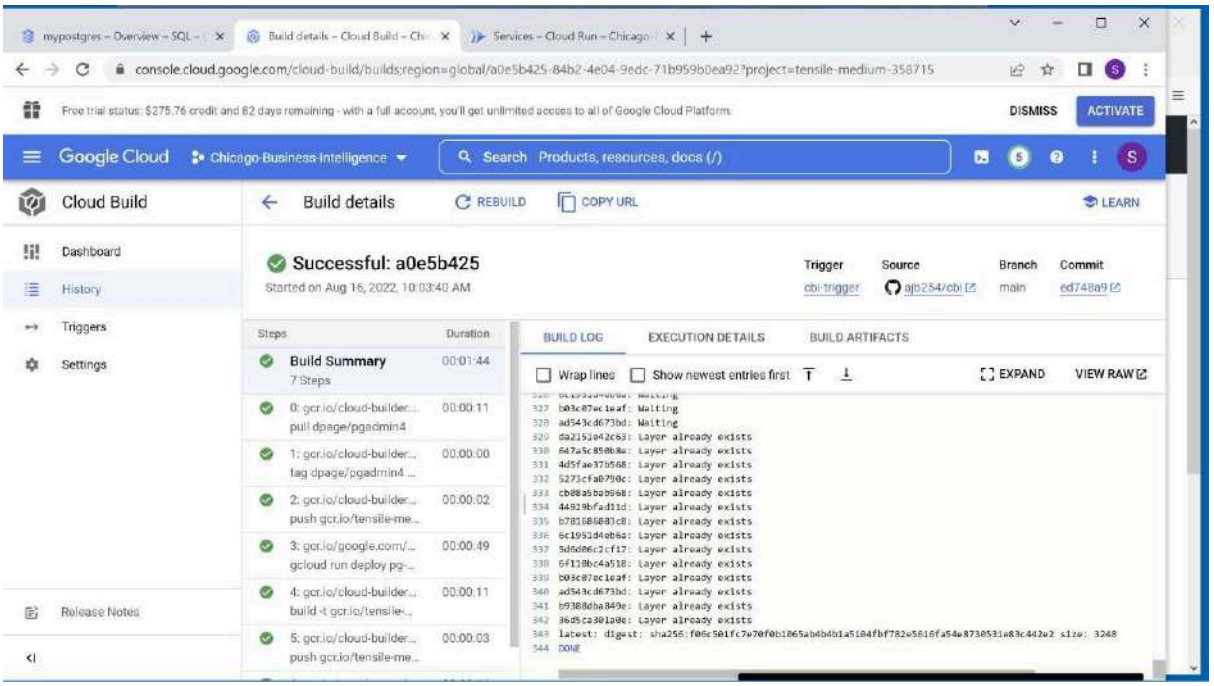


- A build is triggered in cloud build immediately after pushing the code to GitHub.

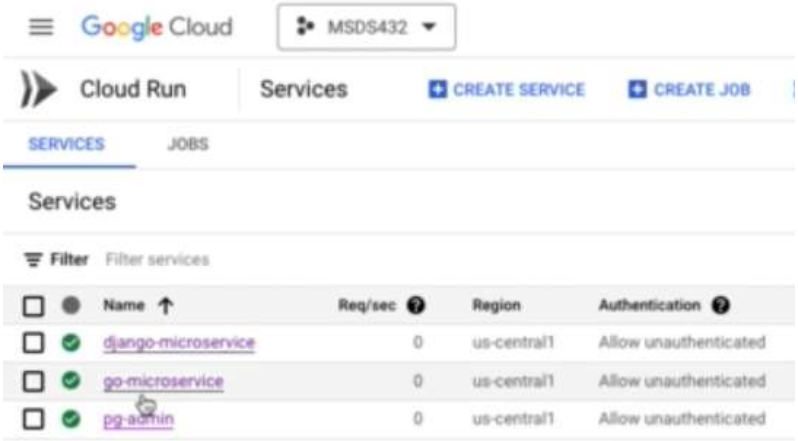


- Wait for the build to be complete. Build logs can be viewed by clicking on the build id.



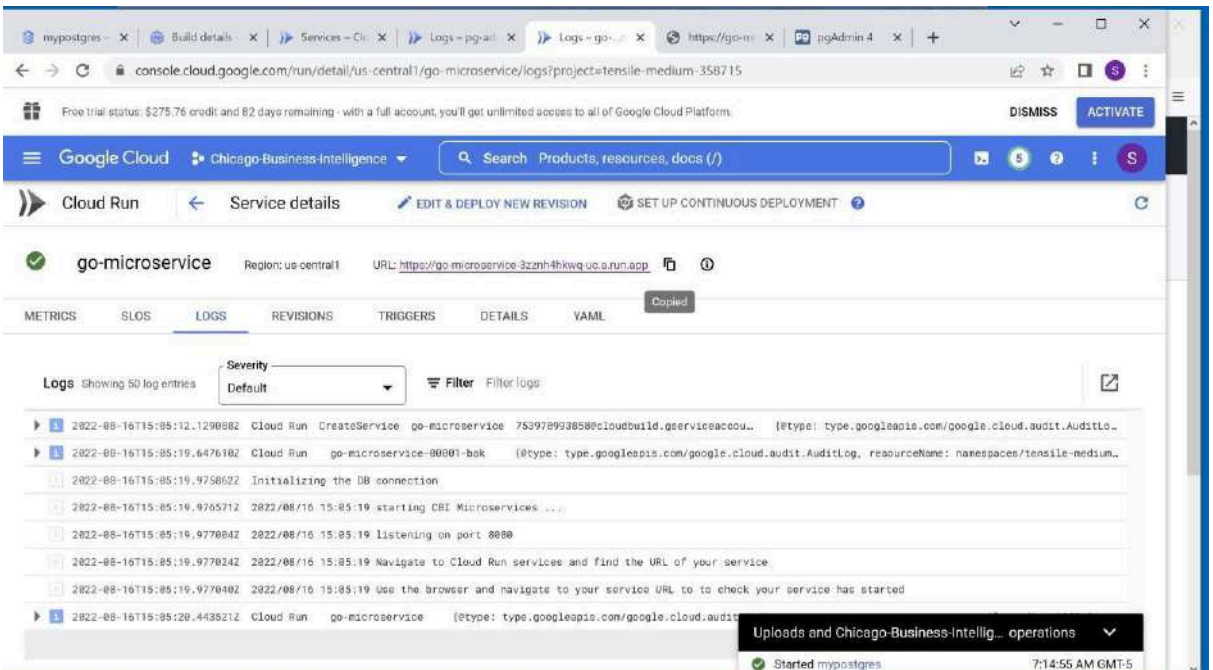


- Go to Cloud Run and Verify you see your services are up and running (green).



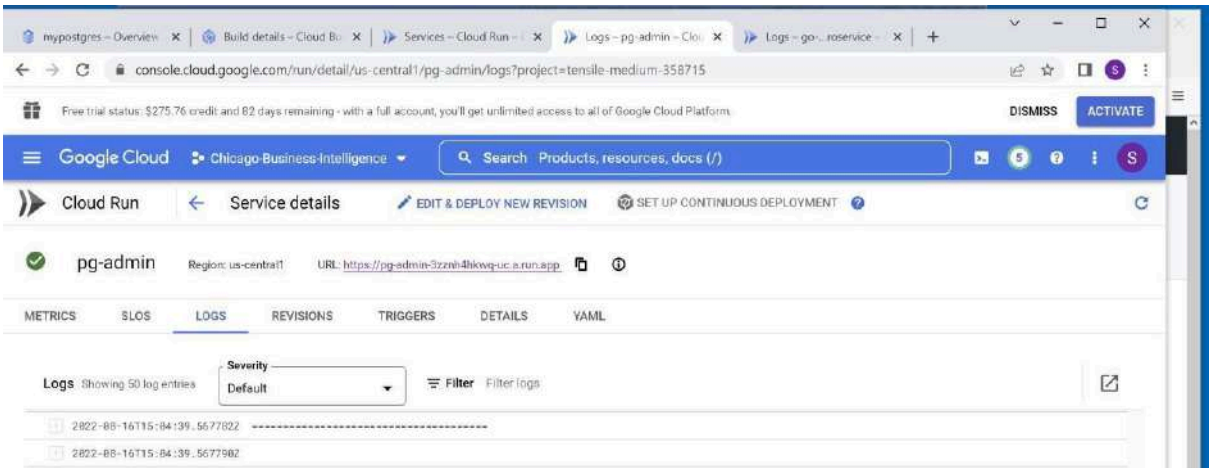
Go Microservice (Backend)

- From Cloud Run, click on the go-microservice app, copy the highlighted URL.
- Open the go-microservice URL in a new browser window and you should see the goroutines for the microservices started.

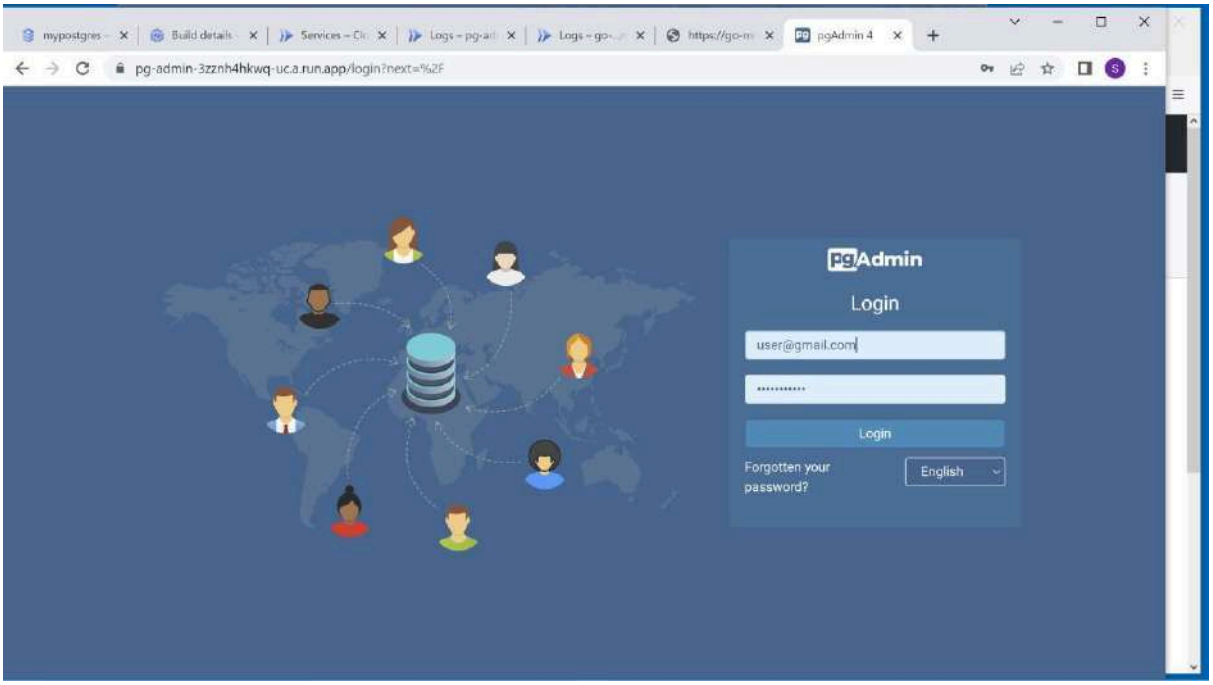


pgAdmin Microservice

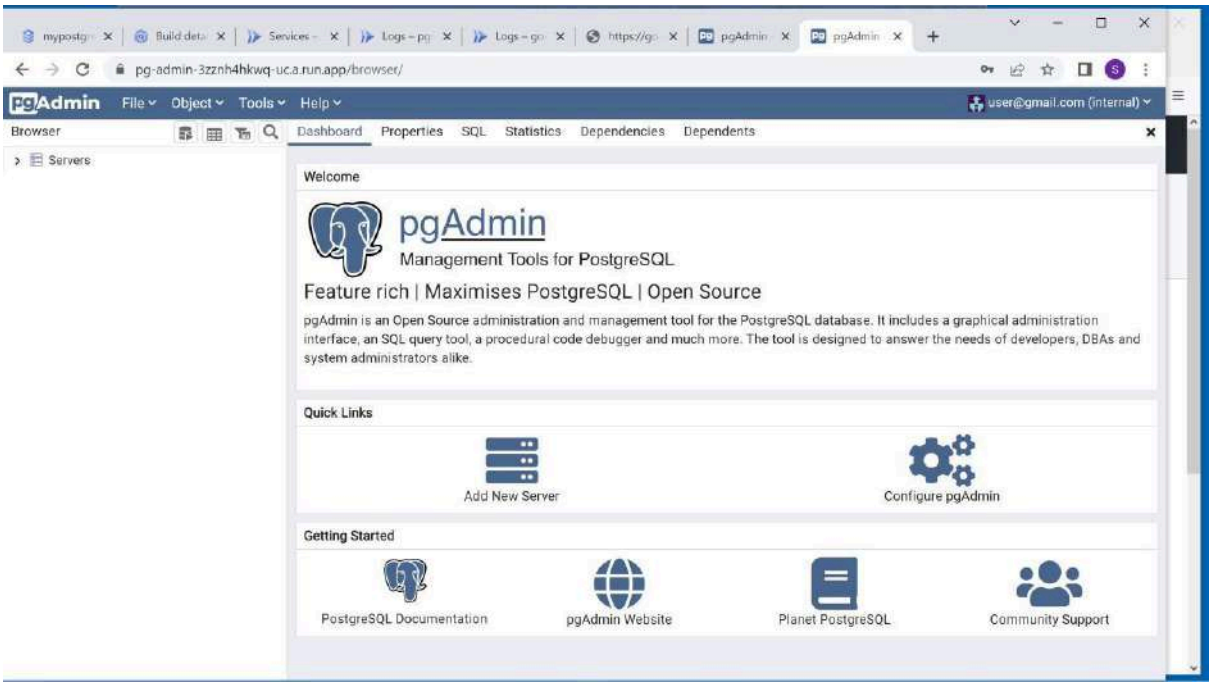
- From Cloud Run, click on pgadmin, copy the highlighted URL.

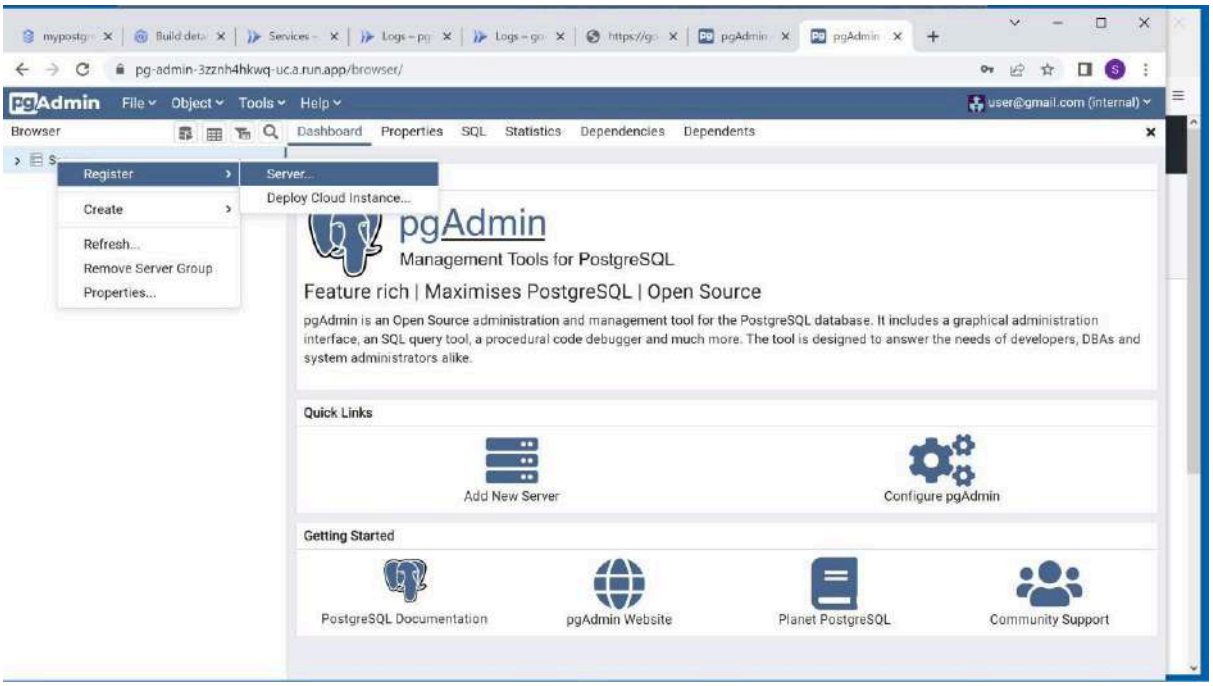


- Open the URL in a Browser and Login to pgadmin to validate that tables are created. Login credentials are in the cloudbuild.yaml file.

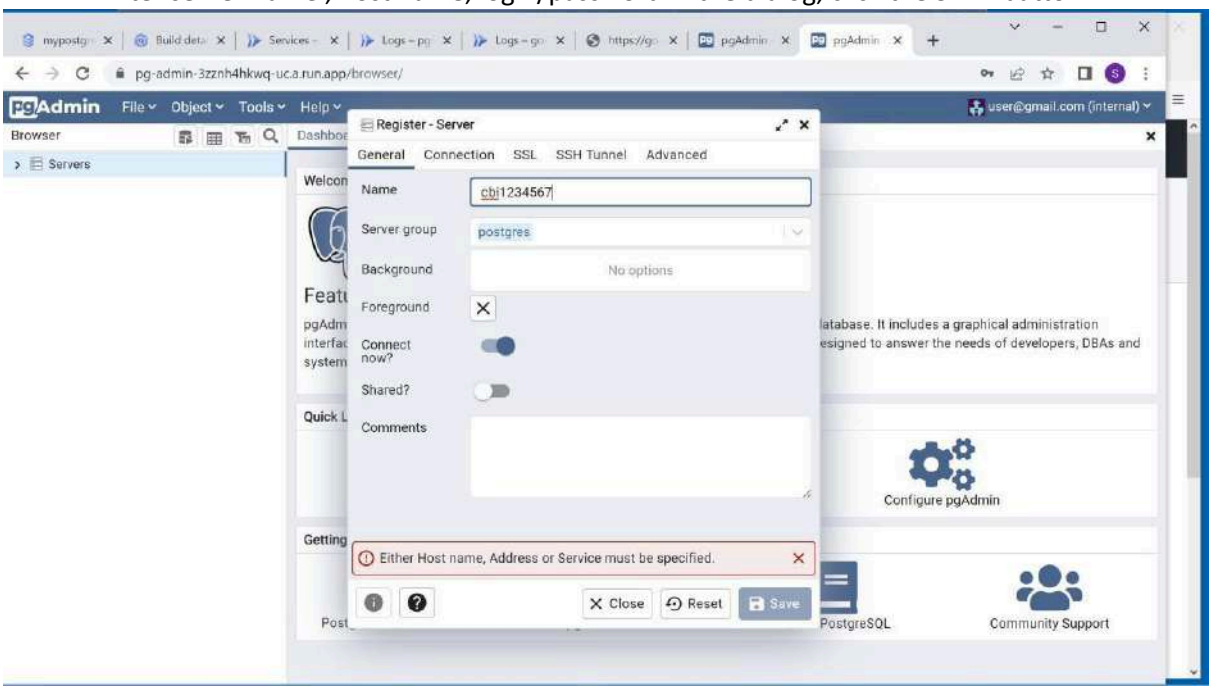


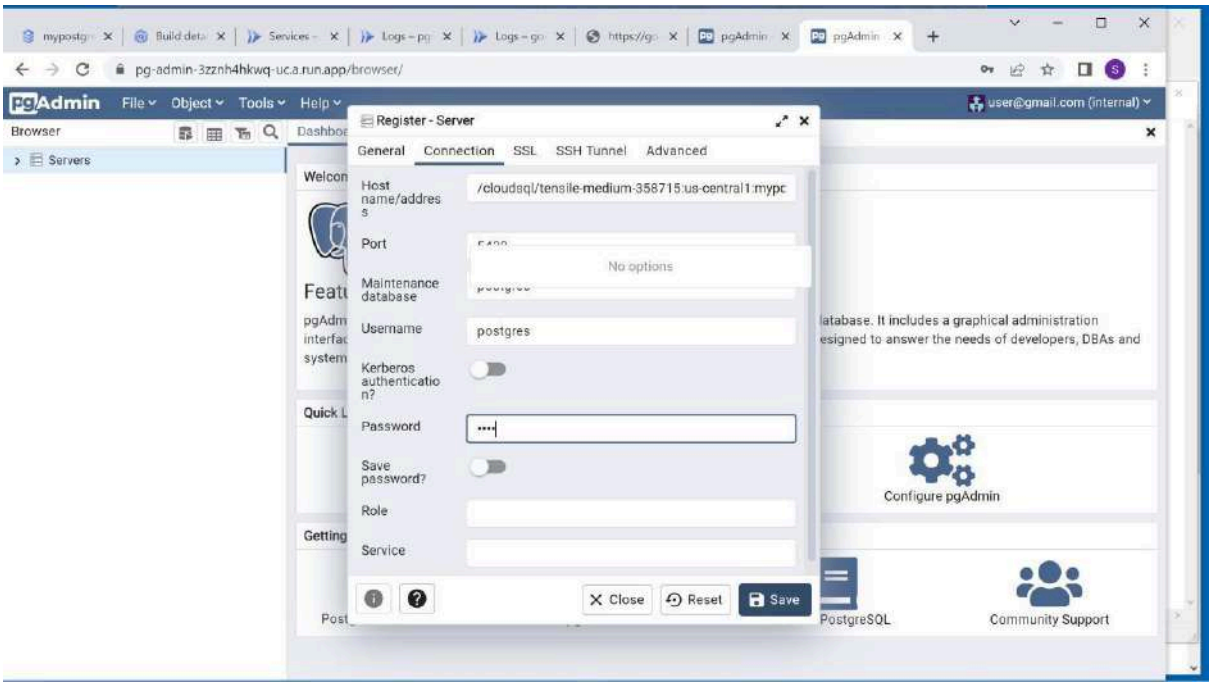
- Add a server.


pg-admin-3zznh4hkwq-uc.a.run.app/browser/
pgAdmin
File
Object
Tools
Help
user@gmail.com (internal)
Browser
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Servers
Welcome
pgAdmin
Management Tools for PostgreSQL
Feature rich | Maximises PostgreSQL | Open Source
pgAdmin is an Open Source administration and management tool for the PostgreSQL database. It includes a graphical administration interface, an SQL query tool, a procedural code debugger and much more. The tool is designed to answer the needs of developers, DBAs and system administrators alike.
Quick Links
Add New Server
Configure pgAdmin
Getting Started
PostgreSQL Documentation
pgAdmin Website
Planet PostgreSQL
Community Support

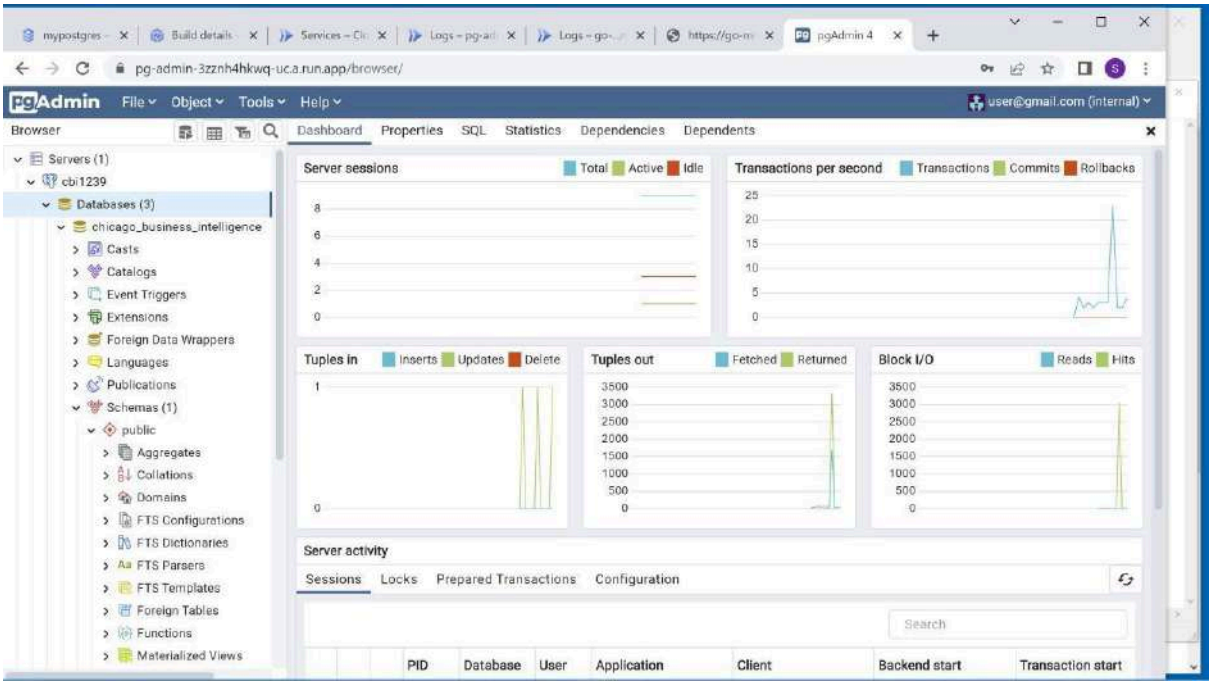

pg-admin-3zznh4hkwq-uc.a.run.app/browser/
pgAdmin
File
Object
Tools
Help
user@gmail.com (internal)
Browser
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Register
Server...
Deploy Cloud Instance...
Create
Refresh...
Remove Server Group
Properties...
pgAdmin
Management Tools for PostgreSQL
Feature rich | Maximises PostgreSQL | Open Source
pgAdmin is an Open Source administration and management tool for the PostgreSQL database. It includes a graphical administration interface, an SQL query tool, a procedural code debugger and much more. The tool is designed to answer the needs of developers, DBAs and system administrators alike.
Quick Links
Add New Server
Configure pgAdmin
Getting Started
PostgreSQL Documentation
pgAdmin Website
Planet PostgreSQL
Community Support

- Enter server name , host name, login/password in the dialog, click the SAVE button.

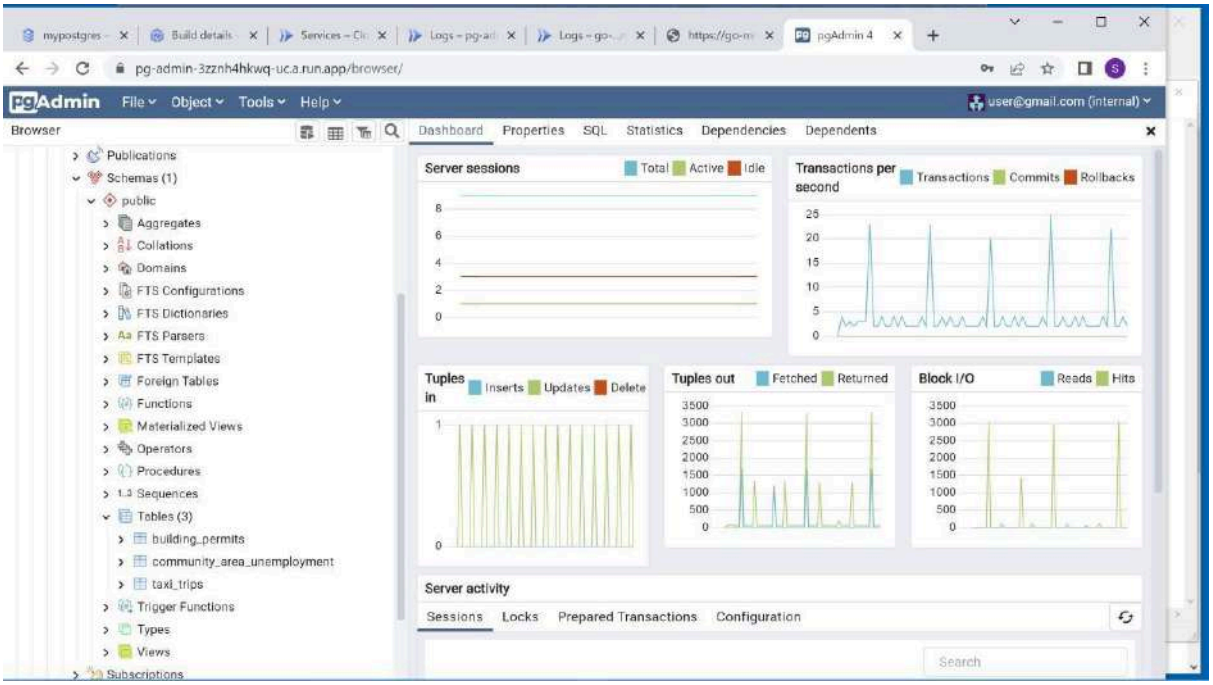




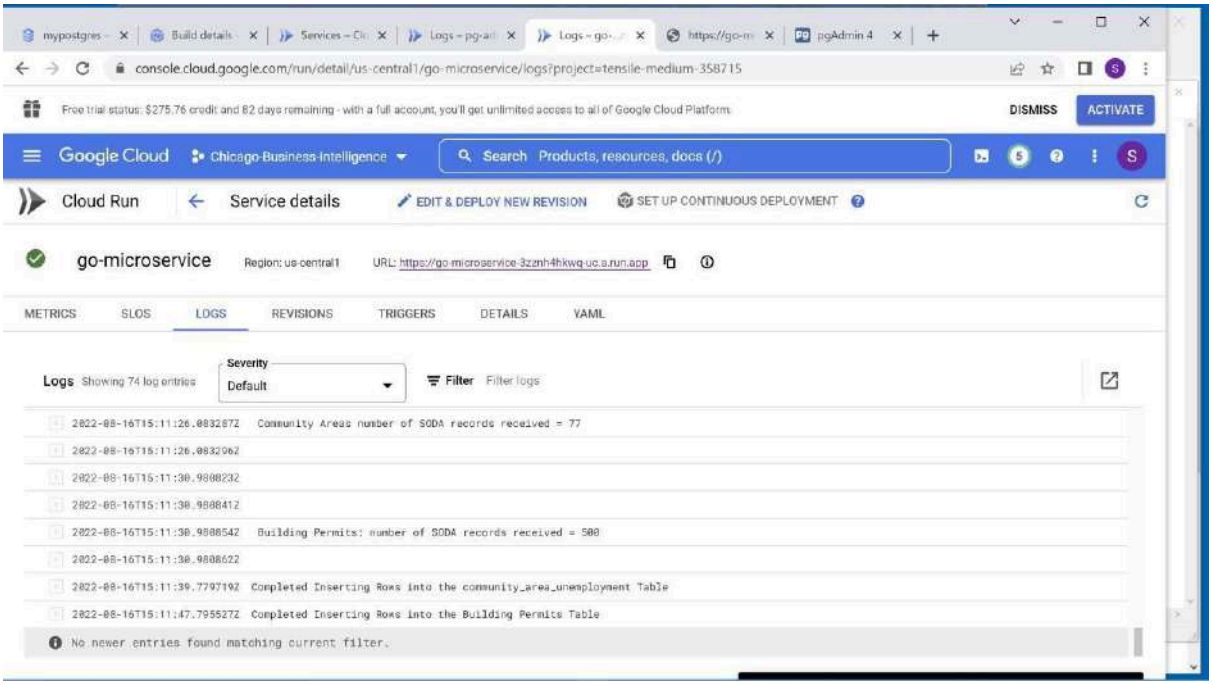
- After you login, click on Chicago_business intelligence.



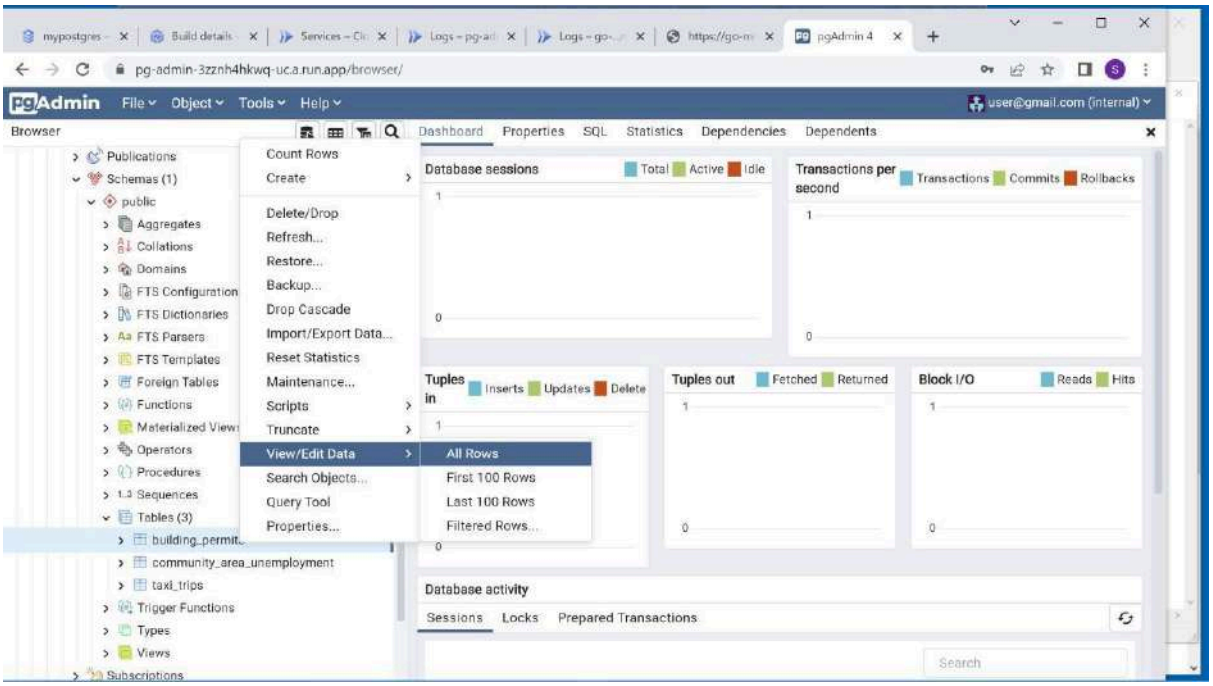
- Click on schemas/tables and verify you see the CBI tables



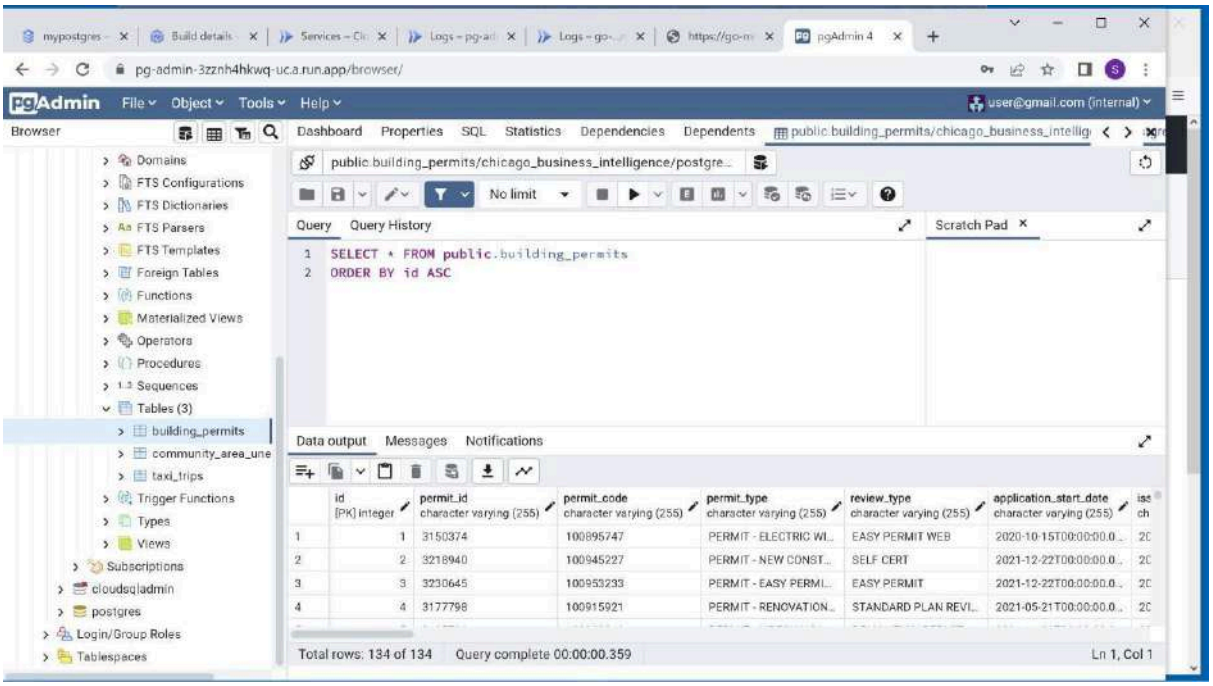
- Go to go-microservices and verify at least one of these tables got rows inserted into it.



- Go back to pgAdmin and select one of these tables and select view all rows:

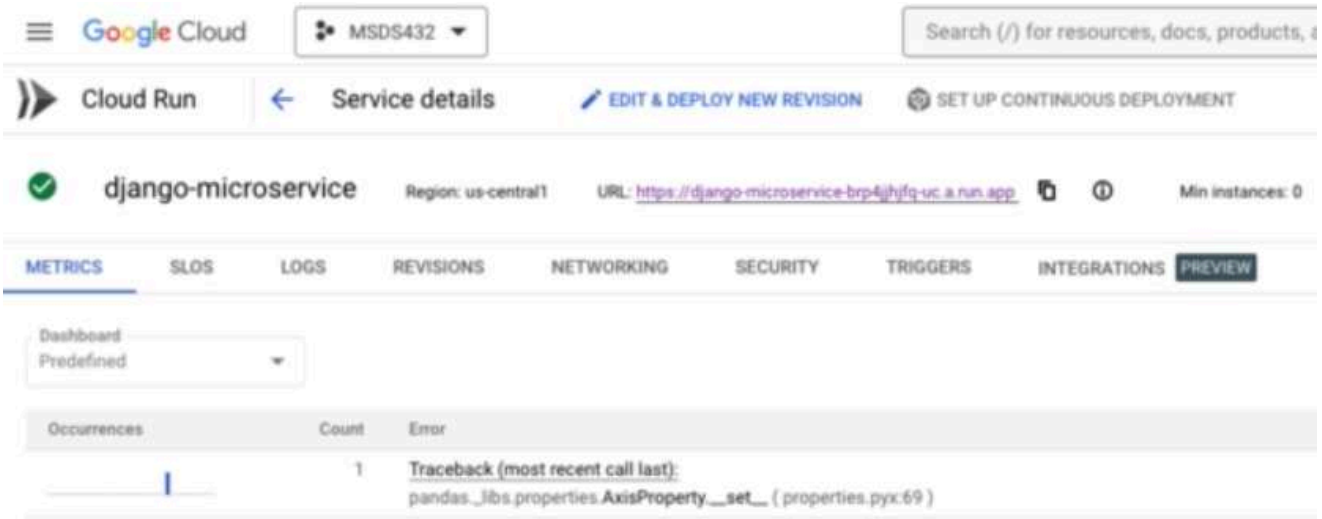


- Go back to pgAdmin and select one of these tables and select view all rows:



Django Microservice (Frontend)

- From Cloud Run, click on the Django Microservice, copy the highlighted URL to access the dashboard.



Welcome to the Chicago Business Intelligence Dashboards

- Taxi Trips From Airport
- Taxi Trips with High CCVI
- Building Fee Per Year
- New Construction in Low Income Zip Codes Per Year